

Finite volume solvers and moving least-squares approximations for the compressible Navier-Stokes equations on unstructured grids

Luis Cueto-Felgueroso, Ignasi Colominas^{*}, Fermín Navarrina
and Manuel Casteleiro

*Group of Numerical Methods in Engineering, GMNI
Dept. of Applied Mathematics, School of Civil Engineering
Universidad de La Coruña
Campus de Elviña, 15071 La Coruña, Spain*

Abstract

This paper introduces the use of Moving Least-Squares (MLS) approximations for the development of high order upwind schemes on unstructured grids, applied to the numerical solution of the compressible Navier-Stokes equations. This mesh-free interpolation technique is designed to reproduce arbitrary functions and their successive derivatives from scattered, pointwise data, which is precisely the case of unstructured-grid finite volume discretizations. The Navier-Stokes solver presented in this study follows the ideas of the generalized Godunov scheme, using Roe's approximate Riemann solver for the inviscid fluxes. Linear, quadratic and cubic polynomial reconstructions are developed using MLS to compute high order derivatives of the field variables. The diffusive fluxes are computed using MLS as a global reconstruction procedure. Various examples of inviscid and viscous flow are presented and discussed.

Key words: Compressible flow, Finite volume method, High-resolution methods, High-order methods, Moving Least-Squares, Unstructured grids.

^{*} Correspondence to: E.T.S. de Ingenieros de Caminos, Canales y Puertos, Universidad de La Coruña, Campus de Elviña, 15071 La Coruña, Spain.
Email address: icolominas@udc.es (Ignasi Colominas).

1 Introduction

An important challenge concerning upwind schemes and unstructured grids is the desire to develop robust and accurate high order methods with a strong multidimensional character. A popular and efficient approach to achieve such objectives is to follow the ideas of the generalized Godunov method [1–3], performing piecewise polynomial reconstructions of the field variables inside each cell, and subsequently using those reconstructed variables as input data for a numerical flux function [3–6].

In practice, the construction of very high order schemes of this kind has been severely limited by the absence of robust approximation techniques, capable of providing accurate estimates of the successive derivatives of the field variables on unstructured grids. Thus, the concept of high-order scheme is most frequently used in the literature in reference to formally second order schemes (piecewise linear reconstruction). Very few attempts to develop higher order reconstructions have been published. One of the most successful of them is the quadratic reconstruction proposed by Barth [7,8], where first order gradients and Hessians are computed separately by means of classical least-squares procedures.

None of the existing approaches provide a general approximation framework, such as that underlying in the concept of *shape functions* in finite element methods. The shape functions represent a very powerful computational feature of finite elements, since the accuracy of the interpolation can be controlled, and they also provide a clear approach to readily extract information about the flow variables within the domain.

This paper proposes the combination of moving least-squares (MLS) approximations and finite volume upwind schemes for Euler and Navier-Stokes computations on unstructured grids. Originally devised for data processing and surface generation [9], the MLS approximation has become very popular among those researchers working in the class of so-called *meshless* or *meshfree* methods, being widely used both in Eulerian and Lagrangian formulations. The approximation power of moving least-squares and reproducing kernel methods has been extensively analyzed both from theoretical and purely numerical approaches [10–14]. This class of approximation methods is particularly well suited for the reconstruction of a given function and its successive derivatives from scattered, pointwise data. This fact suggests the incorporation of MLS approximants into finite volume methods for unstructured grids, whose lack of an underlying interpolation framework is the origin of various shortcomings. We believe this approach somewhat provides a kind of “shape functions” for unstructured-grid finite volume solvers and represents an interesting alternative to the existing approaches.

The Navier-Stokes solver presented in this paper follows the ideas of the generalized Godunov scheme, using Roe’s approximate Riemann solver [15] for the inviscid fluxes. Linear, quadratic and cubic polynomial reconstructions are developed using MLS to compute high order derivatives of the field variables. The diffusive fluxes are computed using MLS as a (centered) global reconstruction procedure.

Although the moving least-squares approximants represent, in essence, a family of global reconstruction operators by themselves, the underlying reconstruction scheme has, in principle, a centered character and, therefore, should not be directly used for the evaluation of the inviscid fluxes at the cell interfaces. However, they are perfectly suitable for the evaluation of the diffusive fluxes in Navier-Stokes computations. Given their centered character and accuracy, MLS approximants also possess nice properties to be used in the development of centered schemes with added dissipation [16]. A current research line pursued by the authors relates to the possibility of using MLS approximants for the direct reconstruction of the convective fluxes, employing adaptive stencils or adaptive kernel functions to provide the necessary upwinding.

The outline of the paper is as follows. Section 2 presents a brief introduction to Moving Least Squares Reproducing Kernel approximation methods. The model equations, numerical formulation and other practical implementation issues are discussed in section 3. Section 4 is devoted to various numerical examples and, finally, section 5 presents the main conclusions of this work.

2 Moving Least Squares Reproducing Kernel approximations

2.1 General formulation.

Let us consider a function $u(\mathbf{x})$ defined in a domain Ω . The basic idea of the Moving Least-Squares (MLS) approach is to approximate $u(\mathbf{x})$, at a given point \mathbf{x} , through a weighted least-squares fitting of $u(\mathbf{x})$ in a neighbourhood of \mathbf{x} , as

$$u(\mathbf{x}) \approx \hat{u}(\mathbf{x}) = \sum_{i=1}^m p_i(\mathbf{x}) \alpha_i(\mathbf{z}) \Big|_{\mathbf{z}=\mathbf{x}} = \mathbf{p}^T(\mathbf{x}) \boldsymbol{\alpha}(\mathbf{z}) \Big|_{\mathbf{z}=\mathbf{x}} \quad (1)$$

where $\mathbf{p}^T(\mathbf{x})$ is an m -dimensional basis of functions (usually polynomials) and $\boldsymbol{\alpha}(\mathbf{z}) \Big|_{\mathbf{z}=\mathbf{x}}$ is a set of parameters to be determined, and such that they minimize

the following error functional

$$J(\boldsymbol{\alpha}(\mathbf{z})|_{\mathbf{z}=\mathbf{x}}) = \int_{\mathbf{y} \in \Omega_{\mathbf{x}}} W(\mathbf{z} - \mathbf{y}, h)|_{\mathbf{z}=\mathbf{x}} [u(\mathbf{y}) - \mathbf{p}^T(\mathbf{y})\boldsymbol{\alpha}(\mathbf{z})|_{\mathbf{z}=\mathbf{x}}]^2 d\Omega_{\mathbf{x}} \quad (2)$$

being $W(\mathbf{z} - \mathbf{y}, h)|_{\mathbf{z}=\mathbf{x}}$ a *kernel* (also smoothing or window function) with compact support (denoted by $\Omega_{\mathbf{x}}$) centered at $\mathbf{z} = \mathbf{x}$. The parameter h , usually called *smoothing length* or *dilatation parameter* in the meshfree literature, is a certain characteristic measure of the size of the support $\Omega_{\mathbf{x}}$ (e.g. kernels with circular supports of radius $2h$). Exponential and spline functions are most frequent kernels.

In numerical computations the global domain Ω is represented by a set of *nodes* or *particles*. The integral in (2) is thus evaluated using those nodes inside $\Omega_{\mathbf{x}}$ as quadrature points. In discrete form, the set of parameters $\boldsymbol{\alpha}$ that minimize the functional J are given by

$$\boldsymbol{\alpha}(\mathbf{z})|_{\mathbf{z}=\mathbf{x}} = \mathbf{M}^{-1}(\mathbf{x})\mathbf{P}_{\Omega_{\mathbf{x}}}\mathbf{W}_{\mathbf{V}}(\mathbf{x})\mathbf{u}_{\Omega_{\mathbf{x}}} \quad (3)$$

where the vector $\mathbf{u}_{\Omega_{\mathbf{x}}}$ contains the pointwise values of the function to be reproduced, $u(\mathbf{x})$, at the $n_{\mathbf{x}}$ nodes inside $\Omega_{\mathbf{x}}$ (figure 1)

$$\mathbf{u}_{\Omega_{\mathbf{x}}} = \left(u(\mathbf{x}_1) \quad u(\mathbf{x}_2) \quad \cdots \quad u(\mathbf{x}_{n_{\mathbf{x}}}) \right)^T \quad (4)$$

The moment matrix \mathbf{M} , which is an $(m \times m)$ matrix, is given by $\mathbf{M}(\mathbf{x}) = \mathbf{P}_{\Omega_{\mathbf{x}}}\mathbf{W}_{\mathbf{V}}(\mathbf{x})\mathbf{P}_{\Omega_{\mathbf{x}}}^T$, and the matrices $\mathbf{P}_{\Omega_{\mathbf{x}}}$ and $\mathbf{W}_{\mathbf{V}}(\mathbf{x})$, whose dimensions are, respectively, $(m \times n_{\mathbf{x}})$ and $(n_{\mathbf{x}} \times n_{\mathbf{x}})$, can be obtained as

$$\mathbf{P}_{\Omega_{\mathbf{x}}} = \left(\mathbf{p}(\mathbf{x}_1) \quad \mathbf{p}(\mathbf{x}_2) \quad \cdots \quad \mathbf{p}(\mathbf{x}_{n_{\mathbf{x}}}) \right) \quad (5)$$

$$\mathbf{W}_{\mathbf{V}}(\mathbf{x}) = \text{diag} \{W_i(\mathbf{x} - \mathbf{x}_i)V_i\}, \quad i = 1, \dots, n_{\mathbf{x}} \quad (6)$$

Complete details can be found in [10,11]. In the above equations, V_i and \mathbf{x}_i denote, respectively, the tributary volume (used as quadrature weight) and coordinates associated to node i . Note that the tributary volumes of the neighbouring nodes are included in matrix $\mathbf{W}_{\mathbf{V}}$, obtaining an MLS version of the Reproducing Kernel Particle Method [10]. Otherwise, we can use \mathbf{W} instead of $\mathbf{W}_{\mathbf{V}}$

$$\mathbf{W}(\mathbf{x}) = \text{diag} \{W_i(\mathbf{x} - \mathbf{x}_i)\}, \quad i = 1, \dots, n_{\mathbf{x}} \quad (7)$$

which corresponds to the classical MLS approximation (in the nodal integration of the functional (2), the same quadrature weight is associated to all

nodes). Introducing (3) in (1), the interpolation structure can be identified as

$$\hat{u}(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{M}^{-1}(\mathbf{x})\mathbf{P}_{\Omega_{\mathbf{x}}}\mathbf{W}(\mathbf{x})\mathbf{u}_{\Omega_{\mathbf{x}}} = \mathbf{N}^T(\mathbf{x})\mathbf{u}_{\Omega_{\mathbf{x}}} = \sum_{j=1}^{n_{\mathbf{x}}} N_j(\mathbf{x})u_j \quad (8)$$

In analogy to finite elements, the approximation was written in terms of the MLS “shape functions”

$$\mathbf{N}^T(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{M}^{-1}(\mathbf{x})\mathbf{P}_{\Omega_{\mathbf{x}}}\mathbf{W}(\mathbf{x}) \quad (9)$$

The functional basis $\mathbf{p}(\mathbf{x})$ is strongly related to the accuracy of the MLS fit. Numerical tests [13] have shown that, for a r^{th} order MLS fit (r^{th} order complete polynomial basis) and general, irregularly spaced points, the nominal order of accuracy for the approximation of a s^{th} order gradient is roughly $(r - s + 1)$. In general, any linear combination of the functions included in the basis is exactly reproduced by the MLS approximation.

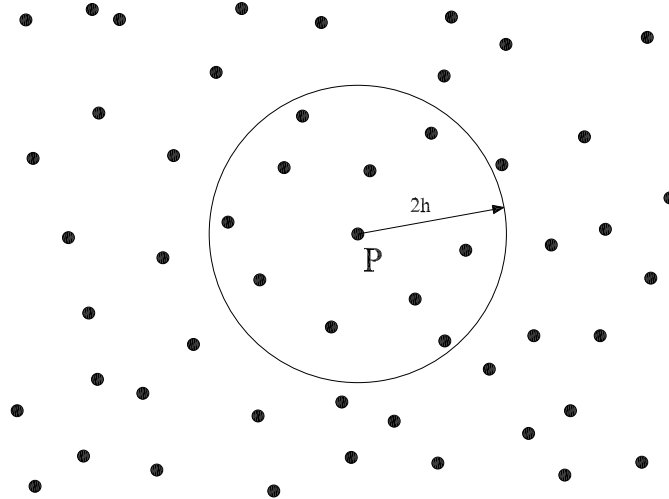


Fig. 1. Meshfree approximation: general scheme. Support for reconstruction at P.

In this study, the following cubic polynomial basis was used

$$\mathbf{p}(\mathbf{x}) = \left(1 \quad x_1 \quad x_2 \quad x_1x_2 \quad x_1^2 \quad x_2^2 \quad x_1^2x_2 \quad x_1x_2^2 \quad x_1^3 \quad x_2^3\right)^T \quad (10)$$

which provides cubic completeness. In the above expression, (x_1, x_2) denotes the cartesian coordinates of \mathbf{x} . To improve the conditioning of the moment matrix, it is most frequent to use scaled and *locally* defined monomials in the basis. Thus, if the shape functions were to be evaluated at a certain point \mathbf{x}_I , the basis would be of the form $\mathbf{p}\left(\frac{\mathbf{x}-\mathbf{x}_I}{h}\right)$, instead of $\mathbf{p}(\mathbf{x})$. With this transformation, the MLS shape functions read

$$\mathbf{N}^T(\mathbf{x}_I) = \mathbf{p}^T(\mathbf{0})\mathbf{C}(\mathbf{x}_I) = \mathbf{p}^T(\mathbf{0})\mathbf{M}^{-1}(\mathbf{x}_I)\mathbf{P}_{\Omega_{\mathbf{x}_I}}\mathbf{W}(\mathbf{x}_I) \quad (11)$$

where $\mathbf{C}(\mathbf{x})$ was defined as

$$\mathbf{C}(\mathbf{x}) = \mathbf{M}^{-1}(\mathbf{x})\mathbf{P}_{\Omega_{\mathbf{x}}}\mathbf{W}(\mathbf{x}) \quad (12)$$

The approximate derivatives of $u(\mathbf{x})$ can be expressed in terms of the derivatives of the MLS shape functions, which are obtained in terms of the derivatives of the polynomial basis $\mathbf{p}(\frac{\mathbf{x}-\mathbf{x}_l}{h})$ and the derivatives of $\mathbf{C}(\mathbf{x})$ [17,16,18]. A wide variety of kernel functions appear in the literature, most of them being spline or exponential functions. In this study we use a cubic spline with radial weighting.

The first order derivatives of the shape functions are computed in this study as full MLS derivatives, whereas second and third order derivatives are *approximated* by the *diffuse* ones. In the diffuse approach, the successive derivatives of $\mathbf{C}(\mathbf{x})$ are neglected. Note that the diffuse derivatives of the shape functions are readily obtained once the matrix $\mathbf{C}(\mathbf{x})$ is computed. It has been shown (see [19] and references therein) that the diffuse derivatives converge at optimal rate to the exact derivatives.

Complete details of the MLS procedure used in this paper can be found in [17,16].

2.2 Computational aspects.

The evaluation of the shape functions at a given point involves a series of matrix operations, the most expensive of them being the inversion of the moment matrix \mathbf{M} . The size of this matrix is $m \times m$, where m is the dimension of the basis $\mathbf{p}(\mathbf{x})$. For the cubic basis (10), for instance, $m = 4$, $m = 10$ and $m = 20$ in one, two and three dimensions, respectively. Note that the size of \mathbf{M} does not depend on the number of neighbours included in the computation of the shape functions.

In order to prevent the matrix \mathbf{M} from being singular or ill-conditioned, the cloud of neighbours should fulfill certain “good neighbourhood” requirements. Thus, if the number of neighbours is less than m (the number of functions in the basis), \mathbf{M} becomes singular, which implies that more than 10 neighbours are needed in 2D computations with the cubic basis. Nevertheless, the approximation could be poor if \mathbf{M} is severely ill-conditioned, so it is convenient to use a number of neighbours greater than the minimum, and with the information coming from as many directions as possible.

2.3 *Moving least-squares, finite volume solvers, unstructured grids.*

This paper proposes the use of MLS approximations to construct high order upwind schemes, through piecewise polynomial reconstructions of the flow variables. The MLS technique was designed to reproduce arbitrary functions and their derivatives from scattered, pointwise data and, therefore, the methodology proposed is suitable for general unstructured-grid FV discretizations, where cell-average variables are associated to the cell centroids.

The proposed scheme follows the ideas of the generalized Godunov method [1,2,7], and requires interpolation capabilities at various stages:

- Reconstruction: for the evaluation of the numerical inviscid fluxes, the field variables are first extrapolated to the interfaces using piecewise polynomial expansions. The construction of such polynomials requires estimating the successive derivatives of the flow variables at the cell centroids.
- Diffusive fluxes: unlike inviscid fluxes, they are computed directly at interfaces. Therefore velocity, temperature, and their corresponding gradients must be evaluated at each edge quadrature point.

All the above interpolation tasks were addressed in this study by means of moving least-squares approximants. Our nodes (in the MLS terminology) will be the cell-centroids and the evaluation points the edge quadrature points and the centroids themselves.

2.4 *Practical computation of the MLS shape functions on unstructured FV discretizations.*

The computation of the MLS shape functions and their derivatives involves two major steps:

- Determination of the “neighbourhood” (cloud of nodes) of the evaluation point; i.e. the set of nodes (centroids) that contribute to the fit.
- Evaluation of the MLS shape functions and their required full/diffuse derivatives, as exposed above.

Once the shape functions and their derivatives have been evaluated at a certain location \mathbf{x} , the flow variables and their successive derivatives are readily computed [17,16]. Note that, using fixed clouds, the MLS shape functions do not change in time and, therefore, they need to be computed only once at the preprocessing phase.

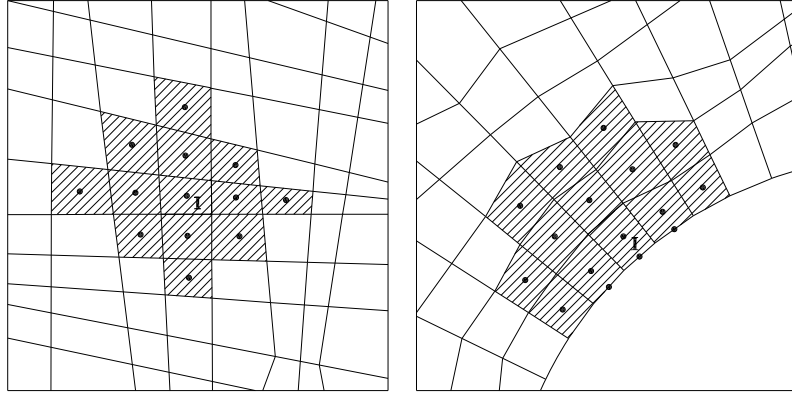


Fig. 2. Typical MLS stencil: centroids.

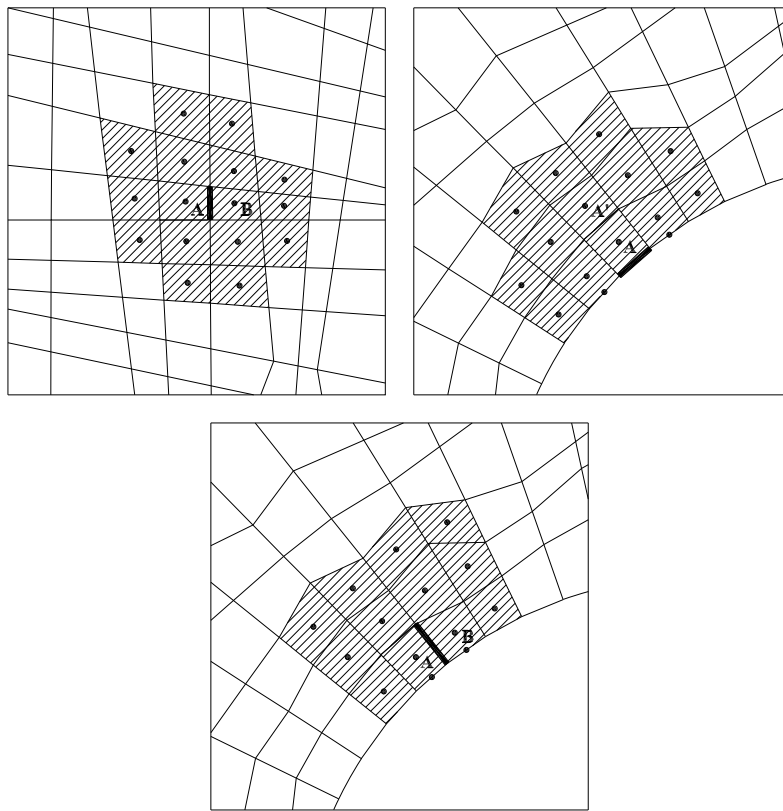


Fig. 3. Typical MLS stencil: quadrature points on edges.

As mentioned before, a key issue concernign MLS approximations in the context of finite volume schemes on unstructured grids is the definition of the cloud of nodes (somewhat the MLS *stencil*) for each evaluation point. The selection process must be suitable for general unstructured grids, and the stencil should be as compact as possible for the sake of computational efficiency and physical meaning. Given that we use a cubic polynomial basis, the stencil's width must cover at least 10 neighbours. In practice a slightly higher number of neighbours would actually be preferred. The stencils chosen here are *fixed*

and defined *a priori*, but ENO-like adaptive clouds are perfectly feasible.

Figures 2 and 3 present the stencils used in this study to compute the MLS shape functions at the centroids and edge quadrature points, respectively. A stronger enforcement of the boundary conditions was achieved through the introduction of a set of “zero area” cells attached to the boundary (an approach analogous to the use of so-called *ghost cells* [6]). Note that the centroids of these boundary cells, i.e. the midpoints of those edges lying on the boundary, have been included in the above stencils.

Once the cloud of neighbour centroids has been determined, the smoothing length h is set to be proportional to the maximum distance between the evaluation point \mathbf{x}_I and its neighbours, as

$$h = k \max(\|\mathbf{x}_j - \mathbf{x}_I\|) \quad (13)$$

Values of k around 0.6–0.7 seem to be adequate (recall that, using radial weighting, the support of the kernel expands over a circle of radius $2h$). Note that a *cell centered* approach with quadrilateral volumes has been adopted in this study. However, the exposed methodology could be extended to any other type of control volumes, provided that suitable stencils are defined.

3 Mathematical model and numerical scheme

3.1 Governing equations.

The compressible Navier-Stokes equations for two-dimensional flow, written in cartesian coordinates and in the absence of source terms, can be cast in conservative form as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial(\mathbf{F}_x - \mathbf{F}_x^V)}{\partial x} + \frac{\partial(\mathbf{F}_y - \mathbf{F}_y^V)}{\partial y} = \mathbf{0} \quad (14)$$

being

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \quad \mathbf{F}_x = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho u H \end{pmatrix}, \quad \mathbf{F}_y = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho v H \end{pmatrix} \quad (15)$$

the conserved variables and inviscid fluxes, respectively, and

$$\mathbf{F}_x^V = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - q_x \end{pmatrix}, \quad \mathbf{F}_y^V = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} - q_y \end{pmatrix} \quad (16)$$

the diffusive fluxes. In the above expressions, ρ denotes density, p pressure and $\mathbf{v} = (u, v)$ is the velocity vector. The total energy and enthalpy are given by

$$\rho E = \rho e + \frac{1}{2}\rho \mathbf{v} \cdot \mathbf{v}, \quad H = E + \frac{p}{\rho} \quad (17)$$

where e is the specific internal energy. The viscous stresses are modelled as

$$\begin{aligned} \tau_{xx} &= 2\mu \frac{\partial u}{\partial x} - \frac{2}{3}\mu \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \\ \tau_{yy} &= 2\mu \frac{\partial v}{\partial y} - \frac{2}{3}\mu \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \\ \tau_{xy} &= \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \end{aligned} \quad (18)$$

where μ is the viscosity. The heat fluxes are assumed to be represented by Fourier's law

$$q_x = -\lambda \frac{\partial T}{\partial x}, \quad q_y = -\lambda \frac{\partial T}{\partial y} \quad (19)$$

where T denotes temperature, $\lambda = c_p \mu / P_r$ is the thermal conductivity, c_p the specific heat at constant temperature ($c_p = 1003.5$ for air) and P_r is the Prandtl number ($P_r = 0.72$ for air). The equation of state and temperature for an ideal gas can be written as

$$p = (\gamma - 1)(\rho E - \frac{1}{2}\rho \mathbf{v} \cdot \mathbf{v}), \quad T = \frac{1}{c_v} \frac{p}{\rho(\gamma - 1)} \quad (20)$$

where c_v is the specific heat at constant volume ($c_v = 716.5$ for air) and $\gamma = \frac{c_p}{c_v}$ is the ratio of specific heats ($\gamma = 1.4$ for air). The speed of sound is given by

$$c = \sqrt{\gamma p / \rho} \quad (21)$$

and the dynamic viscosity μ is assumed to be related to the temperature according to Sutherland's law

$$\mu = \mu_\infty \frac{T + S_0}{T_\infty + S_0} \left(\frac{T}{T_\infty} \right)^{1.5} \quad (22)$$

where μ_∞ and T_∞ denote freestream viscosity and temperature, respectively, and $S_0 = 110.4 \text{ K}$ is an experimental constant [20].

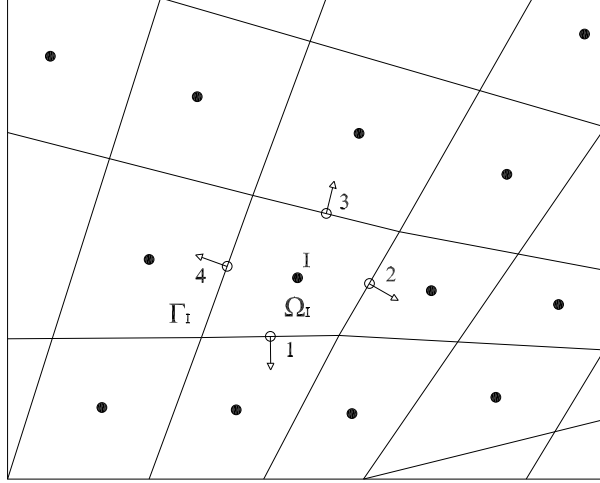


Fig. 4. Cell-centered finite volume discretization.

3.2 Basic finite volume formulation.

The basic finite volume discretization stems from the integral form of the model equations (14) over a control volume Ω_I (figure 4)

$$\int_{\Omega_I} \frac{\partial \mathbf{U}}{\partial t} d\Omega + \int_{\Omega_I} \left(\frac{\partial(\mathbf{F}_x - \mathbf{F}_x^V)}{\partial x} + \frac{\partial(\mathbf{F}_y - \mathbf{F}_y^V)}{\partial y} \right) d\Omega = \mathbf{0} \quad (23)$$

Using the divergence theorem, the above expression reads

$$\int_{\Omega_I} \frac{\partial \mathbf{U}}{\partial t} d\Omega = \int_{\Gamma_I} (\mathcal{F}^V - \mathcal{F}) \cdot \mathbf{n} d\Gamma \quad (24)$$

where $\mathbf{n} = (n_x, n_y)$ is the outward pointing unit normal to the control volume boundary Γ_I , and the definitions

$$\mathcal{F} = (\mathbf{F}_x, \mathbf{F}_y), \quad \mathcal{F}^V = (\mathbf{F}_x^V, \mathbf{F}_y^V) \quad (25)$$

are used for the sake of a more compact presentation. A practical use of (24) involves the development of suitable temporal and spatial discretizations. The former can be addressed by the implementation of standard ODE solvers, whereas the latter gives rise to the particular features of each FV scheme. The idea behind the finite volume method is to discretize the computational domain into a set of non-overlapping control volumes (cells) in which the conservation equations are enforced. In this study a *cell-centered* approach with quadrilateral control volumes was adopted (figure 4), although the use of MLS approximations can be directly extended to more general FV discretizations. From a spatial point of view, the algorithm involves studying the evolution of cell-averaged values of the field variables, and the solution is assumed to be constant within each control volume. Therefore, the underlying spatial representation is that of a piecewise constant flow field. High order schemes are constructed by substituting this piecewise constant representation by a piecewise continuous (usually polynomial) reconstruction of the flow variables inside each cell.

Focusing on control volume I , the discrete version of (24) reads

$$A_I \frac{\partial \mathbf{U}_I}{\partial t} = \sum_{iedge=1}^{nedge_I} \sum_{igau=1}^{ngau_I} [(\mathcal{F}^V - \mathcal{F}) \cdot \mathbf{n}]_{igau} \mathcal{W}_{igau} \quad (26)$$

where A_I is the area of cell I , $nedge_I$ the number of cell edges, $ngau_I$ the number of Gauss quadrature points on each edge, \mathcal{W}_{igau} denotes a quadrature weight and \mathbf{U}_I represents the average value of \mathbf{U} over the cell I .

Given that the field variables are discontinuous across interfaces, the fluxes required to evaluate the expression (26) are not single valued. To overcome this duplicity, a certain *numerical flux* function is introduced. This function returns a single flux value using as input variables the states on each side of the interface. The inviscid numerical flux is frequently designed to include some kind of upwinding, necessary to obtain non-oscillatory and physically meaningful schemes for flow problems. First order schemes use the cell-average values of the variables on each side of the interface as left and right states for the numerical flux, whereas higher order schemes use reconstructed ones, obtained from a certain extrapolation procedure. These ideas are in the basis of the generalized Godunov scheme [1–3], whose implementation involves three major steps:

- Development of piecewise continuous (usually polynomial) reconstructions of the flow variables inside each control volume, using cell-average information from neighbour centroids. The resulting spatial representation is still discontinuous across interfaces. The presence of discontinuities or steep gradients in the solution may require the use of some limiting

strategy.

- Evaluation of fluxes at cell edges. In the case of the inviscid fluxes, the extrapolated left (+) and right (-) states at each edge integration point are used as input data for a numerical flux function (figure 5). In this study, the diffusive fluxes are directly evaluated at the quadrature points using MLS approximation.
- Solution advancement, using appropriate time stepping algorithms.

The following sections elaborate on the particular treatment of the above issues.

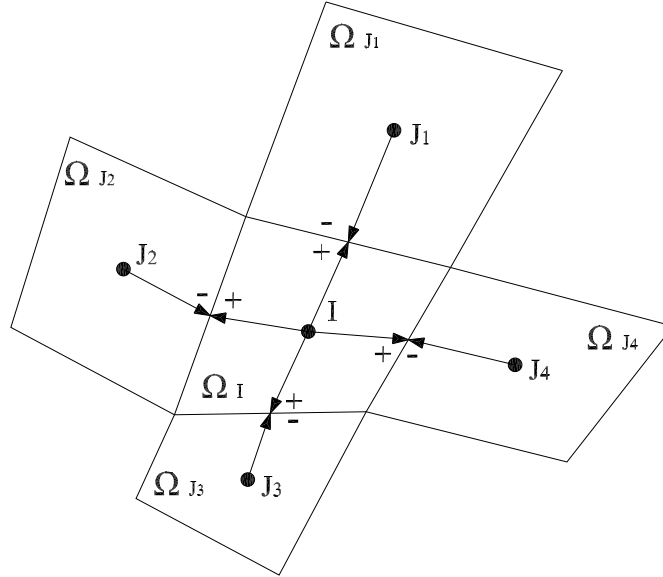


Fig. 5. Extrapolated variables used to evaluate the inviscid fluxes across the interfaces of control volumes Ω_I and $\{\Omega_{J_k}, k = 1, 4\}$.

3.3 Reconstruction and limiting.

This is probably the most complex issue concerning the development of accurate and robust high-order upwind schemes for unstructured grids. Reconstruction is usually addressed by substituting the piecewise constant representation of the basic first order scheme by a piecewise polynomial reconstruction of the field variables inside each control volume, obtained from cell-averaged data. The development of very high order schemes of this kind has been severely limited by the absence of robust approximation techniques, capable of computing accurate estimates of the successive derivatives of the field variables in the context of unstructured grids. Thus, the concept high-order scheme is most frequently used in the literature in reference to formally second-order schemes (piecewise linear reconstruction).

Very few attempts to develop higher order reconstructions have been published. One of the most successful of them is the quadratic reconstruction proposed by Barth [7,8], where first-order gradients and Hessians are computed separately by means of classical least-squares procedures. The most important drawback of this approach is that the least-squares problem posed by the scheme, particularly in the case of the second-order derivatives, is frequently very ill-conditioned.

This section presents linear, quadratic and cubic reconstructions, with first, second and third order derivatives of the field variables computed by means of MLS approximants. These piecewise polynomial reconstructions are obtained using Taylor series expansions; thus, the linear component-wise reconstruction of the variables inside each cell I reads

$$U(\mathbf{x}) = U_I + \nabla U_I \cdot (\mathbf{x} - \mathbf{x}_I) \quad (27)$$

where U_I stands for the cell-averaged (centroid) value, \mathbf{x}_I denotes spatial coordinates of the centroid of the cell and ∇U_I is a cell-centered gradient. This gradient is assumed to be constant on each cell and, therefore, the reconstructed variables are discontinuous across interfaces.

Analogously, the quadratic reconstruction reads

$$U(\mathbf{x}) = U_I + \nabla U_I \cdot (\mathbf{x} - \mathbf{x}_I) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_I)^T \mathbf{H}_I (\mathbf{x} - \mathbf{x}_I) - \frac{1}{2} \left[I_{xx} \frac{\partial^2 U}{\partial x^2} + 2I_{xy} \frac{\partial^2 U}{\partial x \partial y} + I_{yy} \frac{\partial^2 U}{\partial y^2} \right] \quad (28)$$

where \mathbf{H}_I is the centroid Hessian matrix and

$$I_{xx} = \int_{\Omega} r_x^2 d\Omega, \quad I_{xy} = \int_{\Omega} r_x r_y d\Omega, \quad I_{yy} = \int_{\Omega} r_y^2 d\Omega \quad (29)$$

where $r_x = x - x_I$ and $r_y = y - y_I$. The last term in (28) has been introduced to enforce conservation of the mean, i.e.

$$\frac{1}{A_I} \int_{\mathbf{x} \in \Omega_I} U(\mathbf{x}) d\Omega = U_I \quad (30)$$

Note that the incorporation of this term does not reduce the order of the reconstruction given by (28). For steady-state computations we can simply use

$$U(\mathbf{x}) = U_I + \nabla U_I \cdot (\mathbf{x} - \mathbf{x}_I) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_I)^T \mathbf{H}_I (\mathbf{x} - \mathbf{x}_I) \quad (31)$$

Finally, the cubic reconstruction (for steady-state computations) can be written as

$$U(\mathbf{x}) = U_I + \nabla U_I \cdot (\mathbf{x} - \mathbf{x}_I) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_I)^T \mathbf{H}_I (\mathbf{x} - \mathbf{x}_I) + \frac{1}{6} \Delta^2 \mathbf{x}_I^T \mathbf{T}_I (\mathbf{x} - \mathbf{x}_I) \quad (32)$$

where

$$\Delta^2 \mathbf{x}_I^T = \left((x - x_I)^2 \quad (y - y_I)^2 \right), \quad \mathbf{T}_I = \begin{pmatrix} \frac{\partial^3 U_I}{\partial x^3} & 3 \frac{\partial^3 U_I}{\partial x^2 \partial y} \\ 3 \frac{\partial^3 U_I}{\partial x \partial y^2} & \frac{\partial^3 U_I}{\partial y^3} \end{pmatrix} \quad (33)$$

In the case of unlimited reconstructions, the derivatives of the field variables are directly computed at centroids using moving least-squares. Thus, the approximate gradients read

$$\nabla U_I = \sum_{j=1}^{n_{\mathbf{x}_I}} U_j \nabla N_j(\mathbf{x}_I) \quad (34)$$

where the U_j 's stand for variables at the $n_{\mathbf{x}_I}$ "neighbour" (in the sense of the MLS stencil) centroids. The second order derivatives read

$$\begin{aligned} \frac{\partial^2 U_I}{\partial x^2} &= \sum_{j=1}^{n_{\mathbf{x}_I}} U_j \frac{\partial^2 N_j(\mathbf{x}_I)}{\partial x^2} & \frac{\partial^2 U_I}{\partial x \partial y} &= \sum_{j=1}^{n_{\mathbf{x}_I}} U_j \frac{\partial^2 N_j(\mathbf{x}_I)}{\partial x \partial y} \\ \frac{\partial^2 U_I}{\partial y^2} &= \sum_{j=1}^{n_{\mathbf{x}_I}} U_j \frac{\partial^2 N_j(\mathbf{x}_I)}{\partial y^2} \end{aligned} \quad (35)$$

Finally, the third order derivatives are written as

$$\begin{aligned} \frac{\partial^3 U_I}{\partial x^3} &= \sum_{j=1}^{n_{\mathbf{x}_I}} U_j \frac{\partial^3 N_j(\mathbf{x}_I)}{\partial x^3}, & \frac{\partial^3 U_I}{\partial x^2 \partial y} &= \sum_{j=1}^{n_{\mathbf{x}_I}} U_j \frac{\partial^3 N_j(\mathbf{x}_I)}{\partial x^2 \partial y} \\ \frac{\partial^3 U_I}{\partial x \partial y^2} &= \sum_{j=1}^{n_{\mathbf{x}_I}} U_j \frac{\partial^3 N_j(\mathbf{x}_I)}{\partial x \partial y^2}, & \frac{\partial^3 U_I}{\partial y^3} &= \sum_{j=1}^{n_{\mathbf{x}_I}} U_j \frac{\partial^3 N_j(\mathbf{x}_I)}{\partial y^3} \end{aligned} \quad (36)$$

In this study, the first order derivatives were computed as full MLS derivatives, whereas the second and third order derivatives are *approximated* by the diffuse ones.

In the presence of shocks, some limiting procedure is applied to the above derivatives. The choice of adequate multidimensional limiters is critical in order to achieve accurate and non-oscillatory shock-capturing algorithms.

3.3.1 Monotonicity enforcement.

Barth and Jespersen [3] have proposed an extension of Van Leer’s scheme [21] which is suitable for unstructured grids. The basic idea is to enforce “*monotonicity*” in the reconstructed solution. In this context, monotonicity implies that no new extrema are created by the reconstruction process [3]. The enforcement is local, in the sense that only certain neighbour cells are considered for the “no new extrema” criterion.

Recall the piecewise linear reconstruction $U(\mathbf{x})_I$ of a variable U inside a certain cell I

$$U(\mathbf{x})_I = U_I + \nabla U_I \cdot (\mathbf{x} - \mathbf{x}_I) \quad (37)$$

and consider a *limited* version of this reconstruction, as

$$U(\mathbf{x})_I = U_I + \Phi_I \nabla U_I \cdot (\mathbf{x} - \mathbf{x}_I) \quad (38)$$

where Φ_I is a slope limiter ($0 \leq \Phi_I \leq 1$) such that the reconstruction (38) satisfies

$$U^{\min} \leq U(\mathbf{x})_I \leq U^{\max} \quad (39)$$

being

$$U^{\min} = \min_{j \in \mathcal{A}_I} (U_j), \quad U^{\max} = \max_{j \in \mathcal{A}_I} (U_j) \quad (40)$$

where \mathcal{A}_I is the set of “neighbour” cells. In practice, the restriction (39) is only enforced at the quadrature points on the edges of cell I ; thus, for each quadrature point q , its associated slope limiter Φ_I^q is computed in terms of the unlimited extrapolated value U_I^q , as

$$\Phi_I^q = \begin{cases} \min \left(1, \frac{U^{\max} - U_I}{U_I^q - U_I} \right) & U_I^q - U_I > 0 \\ \min \left(1, \frac{U^{\min} - U_I}{U_I^q - U_I} \right) & U_I^q - U_I < 0 \\ 1 & U_I^q - U_I = 0 \end{cases} \quad (41)$$

and, finally,

$$\Phi_I = \min_q(\Phi_I^q) \quad (42)$$

In the case of the quadratic reconstruction (31), a similar limiting strategy can be written as

$$U(\mathbf{x}) = U_I + \Phi_I \left(\nabla U_I \cdot (\mathbf{x} - \mathbf{x}_I) + \frac{1}{2} (\mathbf{x} - \mathbf{x}_I)^T \mathbf{H}_I (\mathbf{x} - \mathbf{x}_I) \right) \quad (43)$$

where the limiter Φ_I is obtained following the same procedure exposed above for the linear case.

In this study the neighbourhood to determine the extremum values U^{\min} and U^{\max} is comprised of the reconstruction cell I and its first order neighbours (figure 6–A). In the following, the above limiter will be referred to as “*BJ limiter*”.

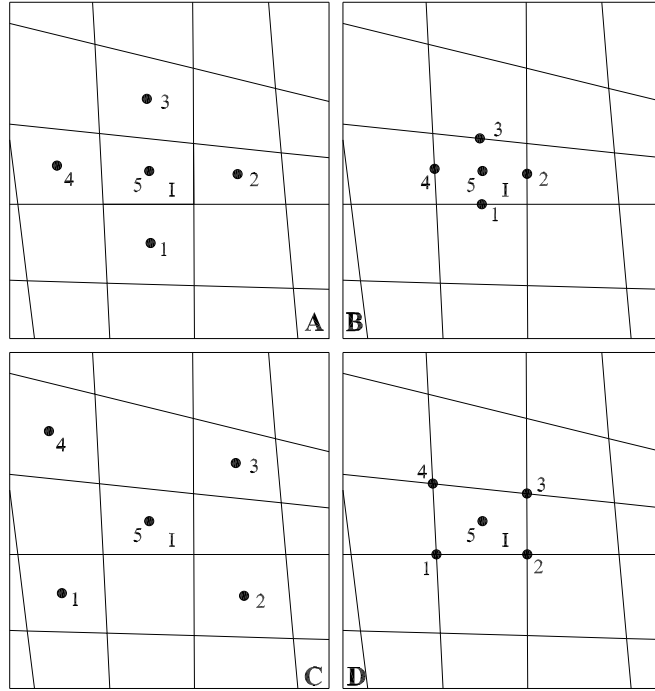


Fig. 6. Neighbourhoods for the limiting of the reconstruction inside cell I .

3.3.2 Averaged derivatives.

This section presents a general strategy to obtain limited gradients and hessian matrices. Thus, the limited gradient associated to a certain cell I , ∇U_I is

obtained as a weighted average of a series of *representative* gradients, as

$$\nabla U_I = \sum_{k=1}^N \omega_k \nabla U_k \quad (44)$$

where $\{\nabla U_k, \quad k = 1, \dots, N\}$ is a set of unlimited gradients, used as a basis to construct the limited one. In an approach similar to that exposed in [6], the weights $\{\omega_k, \quad k = 1, \dots, N\}$ are given by

$$\omega_k (g_1, g_2, \dots, g_N) = \frac{\prod_{i \neq k}^N g_i + \epsilon^{N-1}}{\sum_{j=1}^N \left(\prod_{i \neq j}^N g_i \right) + N \epsilon^{N-1}} \quad k = 1, \dots, N \quad (45)$$

where $\{g_i, \quad i = 1, \dots, N\}$ are functions of the unlimited gradients (in this study, $g_i = \|\nabla U_i\|^2$) and ϵ is a small number, introduced to avoid division by zero. The hessian matrices will also be limited following these ideas but, in this case, the functions g_i read

$$g_i = \left(\frac{\partial^2 U_i}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 U_i}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 U_i}{\partial y^2} \right)^2 \quad i = 1, \dots, N \quad (46)$$

Some existing limiters could be considered to be included in this family. Van Rosendale [22] has proposed an extension to three gradients of Van Albada's limiter [23]. This limiter was used on unstructured triangular grids and its general structure is that of (44) with $N = 3$. The representative gradients are evaluated at the three *vertices* of the cell. Jawahar and Kamath [6] proposed a limiter with $N = 3$, with averaged gradients computed from the unlimited gradients evaluated at the *centroids* of the adjacent cells on triangular meshes. Furthermore, the denominators in (45) are slightly different in this case.

For quadrilateral cells we propose a limiter based on (44)–(45) with $N = 5$; i.e. the limited derivatives are obtained as a weighted average of five unlimited derivatives. Figure 6 presents four suitable configurations to determine such *representative derivatives*. In this study only the configuration given by 6–A will be considered. In the following, the above limiter will be referred to as “*PC5 limiter*”.

3.4 Numerical convective fluxes.

The numerical inviscid fluxes in (26) are obtained using Roe's flux difference splitting [15]. For this purpose, left (\mathbf{U}^+) and right (\mathbf{U}^-) states are defined on

each face. The numerical flux is then computed as

$$(\mathbf{F}_x, \mathbf{F}_y) \cdot \mathbf{n} = \frac{1}{2} \left[(\mathbf{F}_x(\mathbf{U}^+), \mathbf{F}_y(\mathbf{U}^+)) + (\mathbf{F}_x(\mathbf{U}^-), \mathbf{F}_y(\mathbf{U}^-)) \right] \cdot \mathbf{n} - \frac{1}{2} \sum_{k=1}^3 \tilde{\alpha}_k |\tilde{\lambda}_k| \tilde{\mathbf{r}}_k \quad (47)$$

where $\{\tilde{\lambda}_k, k = 1, 4\}$ and $\{\tilde{\mathbf{r}}_k, k = 1, 4\}$ are, respectively, the eigenvalues and eigenvectors of the approximate jacobian $\tilde{\mathbf{J}}(\mathbf{U}^+, \mathbf{U}^-)$

$$\tilde{\lambda}_1 = \tilde{\mathbf{v}} \cdot \mathbf{n} - \tilde{c}, \quad \tilde{\lambda}_2 = \tilde{\lambda}_3 = \tilde{\mathbf{v}} \cdot \mathbf{n}, \quad \tilde{\lambda}_4 = \tilde{\mathbf{v}} \cdot \mathbf{n} + \tilde{c} \quad (48)$$

$$(\tilde{\mathbf{r}}_1 \tilde{\mathbf{r}}_2 \tilde{\mathbf{r}}_3 \tilde{\mathbf{r}}_4) = \begin{pmatrix} 1 & 0 & 1 & 0 \\ \tilde{u} - \tilde{c}n_x & -\tilde{c}n_y & \tilde{u} & \tilde{u} + \tilde{c}n_x \\ \tilde{v} - \tilde{c}n_y & \tilde{c}n_x & \tilde{v} & \tilde{v} + \tilde{c}n_y \\ \tilde{H} - \tilde{c} \tilde{\mathbf{v}} \cdot \mathbf{n} & \tilde{c}(\tilde{v}n_x - \tilde{u}n_y) & \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2) & \tilde{H} + \tilde{c} \tilde{\mathbf{v}} \cdot \mathbf{n} \end{pmatrix} \quad (49)$$

and the corresponding wave strengths $\{\tilde{\alpha}_k, k = 1, 4\}$

$$\begin{aligned} \tilde{\alpha}_1 &= \frac{1}{2\tilde{c}^2} [\Delta(p) - \tilde{\rho}\tilde{c}(\Delta(u)n_x + \Delta(v)n_y)] \\ \tilde{\alpha}_2 &= \frac{\tilde{\rho}}{\tilde{c}} [\Delta(v)n_x - \Delta(u)n_y] \\ \tilde{\alpha}_3 &= -\frac{1}{\tilde{c}^2} [\Delta(p) - \tilde{c}^2\Delta(\rho)] \\ \tilde{\alpha}_4 &= \frac{1}{2\tilde{c}^2} [\Delta(p) + \tilde{\rho}\tilde{c}(\Delta(u)n_x + \Delta(v)n_y)] \end{aligned} \quad (50)$$

where $\Delta(\cdot) = (\cdot)^- - (\cdot)^+$, $\mathbf{n} = (n_x, n_y)$ is the outward pointing unit normal to the interface, and the Roe-average values $\tilde{\mathbf{v}} = (\tilde{u}, \tilde{v})$ and \tilde{H} (computed using \mathbf{U}^+ and \mathbf{U}^-) are defined as

$$\tilde{u} = \frac{u^+\sqrt{\rho^+} + u^-\sqrt{\rho^-}}{\sqrt{\rho^+} + \sqrt{\rho^-}} \quad \tilde{v} = \frac{v^+\sqrt{\rho^+} + v^-\sqrt{\rho^-}}{\sqrt{\rho^+} + \sqrt{\rho^-}} \quad \tilde{H} = \frac{H^+\sqrt{\rho^+} + H^-\sqrt{\rho^-}}{\sqrt{\rho^+} + \sqrt{\rho^-}} \quad (51)$$

On the other hand, the average values $\tilde{\rho}$ and \tilde{c} are computed as

$$\tilde{\rho} = \sqrt{\rho^+\rho^-} \quad \tilde{c}^2 = (\gamma - 1) \left[\tilde{H} - \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2) \right] \quad (52)$$

A first order scheme is obtained by setting \mathbf{U}^+ and \mathbf{U}^- to be the cell-averaged variables at the left and right control volumes. Although first order schemes often provide valuable information for the engineering practice, their accuracy

is severely undermined by an excess of numerical dissipation. The so-called “higher order schemes” correspond to using extrapolated left and right states (figure 5), obtained through a reconstruction process, as exposed in section 3.3.

3.5 Diffusive fluxes.

Given that no upwinding is necessary in the case of the diffusive fluxes, we use the MLS approximants as a global (centered) reconstruction procedure to evaluate the diffusive fluxes at the quadrature points on the edges. Recall that the evaluation of the viscous stresses and heat fluxes requires interpolating the velocity vector $\mathbf{v} = (u, v)$, temperature T , and their corresponding gradients, $\nabla\mathbf{v}$ and ∇T , at each quadrature point \mathbf{x}_{iq} . Using MLS approximation, these entities are readily computed as

$$\mathbf{v}_{iq} = \sum_{j=1}^{n_{iq}} \mathbf{v}_j N_j(\mathbf{x}_{iq}), \quad T_{iq} = \sum_{j=1}^{n_{iq}} T_j N_j(\mathbf{x}_{iq}) \quad (53)$$

and

$$\nabla\mathbf{v}_{iq} = \sum_{j=1}^{n_{iq}} \mathbf{v}_j \otimes \nabla N_j(\mathbf{x}_{iq}), \quad \nabla T_{iq} = \sum_{j=1}^{n_{iq}} T_j \nabla N_j(\mathbf{x}_{iq}) \quad (54)$$

where n_{iq} is the number of neighbour centroids (in the sense of the MLS stencil). Once the above information has been interpolated, the diffusive fluxes can be computed, according to (16), and introduced in (26).

3.6 Flux integration.

It is convenient to combine high order polynomial reconstructions with high order quadratures at cell edges. In this study one integration point (the mid-point) was used in the case of linear reconstruction, whereas two and three Gauss points were respectively used in the case of quadratic and cubic reconstructions.

3.7 Time integration.

We use the third order TVD-Runge-Kutta algorithm proposed by Shu and Osher [24]. Given the field variables U^n at the previous time step n , the al-

gorithm proceeds in three stages to obtain the updated field variables U^{n+1} , as

$$\begin{aligned} U^1 &= U^n + \Delta t L(U^n) \\ U^2 &= \frac{3}{4}U^n + \frac{1}{4}U^1 + \frac{1}{4}\Delta t L(U^1) \\ U^{n+1} &= \frac{1}{3}U^n + \frac{2}{3}U^2 + \frac{2}{3}\Delta t L(U^2) \end{aligned} \quad (55)$$

where the operator $L(\cdot)$, which represents the time derivative given by (26), reads

$$L(\mathbf{U}) = \frac{1}{A} \sum_{iedge=1}^{nedge} \sum_{igau=1}^{ngau} [(\mathcal{F}^V - \mathcal{F}) \cdot \mathbf{n}]_{igau} \mathcal{W}_{igau} \quad (56)$$

4 Numerical results

This section presents various numerical examples of flow around a NACA 0012 aerofoil, aimed at a practical assessment of the performance of the proposed methodology. The quadrilateral unstructured grids were generated using the code GEN4U, based on the formulation proposed by Sarrate and Huerta [25].

4.1 Inviscid flow.

4.1.1 A subsonic test.

The MLS approximants provide a general and robust framework to develop high order polynomial reconstructions for upwind FV schemes on unstructured grids, thus enhancing the accuracy of the scheme without introducing additional degrees of freedom. This section analyzes the effect of increasing the order of the reconstruction on a simple subsonic test case.

The problem set up corresponds to a subsonic flow around a NACA 0012 aerofoil. The freestream flow is given by $M = 0.63$, $\alpha = 2^\circ$. The computational grid is rather coarse (figure 7, 5322 cells). Given the poor mesh resolution near the leading and trailing edges, the inherent dissipation associated to each reconstruction becomes clear through the inspection of the Mach number isolines.

Figure 8 shows the Mach number isolines obtained by using linear (A), quadratic (B) and cubic (C and D) reconstructions. The inviscid fluxes have been inte-

grated using one, two and either two (C) or three (D) Gauss points per edge, for the linear, quadratic and cubic reconstructions, respectively.

The solution provided by the linear reconstruction clearly shows the anomalous pseudoviscous behaviour of the Mach number contours near the surface. The entropy layer is dramatically reduced by the increase of the order of the reconstruction. Figure 9 presents a close-up view of the Mach number contours near the surface of the aerofoil.

The maximum entropy production reduces from $\Delta S_{\max} = 0.03336$ (linear reconstruction) to $\Delta S_{\max} = 0.00772$ (cubic reconstruction), where S is given by

$$S = \ln \left(\frac{h^{\frac{\gamma}{\gamma-1}}}{p} \right) \quad h = \gamma \left(E - \frac{1}{2}(u^2 + v^2) \right) \quad (57)$$

Figure 10 plots a comparison of the computed surface pressure coefficient C_p distribution near the leading edge of the aerofoil, which confirms the superior performance of the higher order reconstructions.

4.1.2 Two transonic examples.

A finer grid (figure 11, 12243 cells) has been used to solve to transonic test cases: I) $M = 0.8$, $\alpha = 1.25^\circ$, and II) $M = 0.85$, $\alpha = 1^\circ$. Figures 12 and 13 show the results for test cases I and II, respectively, using quadratic reconstruction and either the BJ or the PC5 limiter: Mach number isolines, pressure isolines and surface pressure coefficient C_p distribution. Both limiters provide sharp shock-capturing (one interior cell) and clear slip lines, although the PC5 limiter appears to be more dissipative.

4.2 Viscous flow.

For Navier-Stokes computations more cells have been clustered around the aerofoil, to give a final grid of 15111 quadrilaterals. Three test cases have been considered: III) $M = 0.8$, $\alpha = 10^\circ$, $Re = 500$, IV) $M = 2$, $\alpha = 0^\circ$, $Re = 2000$ and V) $M = 0.3$, $\alpha = 30^\circ$, $Re = 3000$. Only test case IV requires the use of limiters, in order to deal with the detached bow shock.

Figure 14 shows Mach number isolines, streamline patterns and surface pressure C_p and skin friction C_f coefficients for test case III. Figure 15 shows Mach number isolines and surface pressure and skin friction coefficients for test case

IV. Both problems have been solved using quadratic reconstruction. For test case IV the PC5 limiter has been used.

Under the angle of attack considered in test case V, the laminar flow becomes unsteady, and no steady-state solution is reached. Instead, the flow pattern is characterized by a periodical vortex shedding process. Figure 16 plots the streamline patterns and Mach number field, corresponding to four non-dimensional time values, computed as $t^* = V_\infty t/c$, where V_∞ is the freestream velocity and c is the chord length. The results are in good agreement with those reported in [26], obtained with a structured-grid solver.

5 Conclusions and future developments

This paper explores the approximation power of Moving Least-Squares approximations in the context of high order upwind schemes on unstructured grids. Applications focus on the compressible Navier-Stokes equations and flow around aerofoils. This meshfree interpolation technique is designed to reproduce arbitrary functions and their successive derivatives from scattered, pointwise data, which is precisely the case of unstructured-grid finite volume discretizations.

The Navier-Stokes solver presented here follows the ideas of the generalized Godunov scheme, using Roe's approximate Riemann solver for the inviscid fluxes. Linear, quadratic and cubic polynomial reconstructions are developed using MLS to compute high order derivatives of the field variables. The diffusive fluxes are computed using MLS as a (centered) global reconstruction procedure. Numerical examples of inviscid and viscous flow around a NACA0012 aerofoil demonstrate that the proposed methodology represents an interesting alternative to the existing procedures.

The results obtained so far have encouraged the quest for further developments in the combination of powerful meshfree approximation techniques and finite volume solvers: ENO-like adaptive stencils or adaptive kernels for reconstruction and centered schemes with added dissipation. An important research line currently pursued by the authors is concerned to the assessment of moving least-squares for highly stretched meshes. This kind of mesh structure is typical of high- Re flows and anisotropic adaptive procedures. In these cases, computing accurate approximations of the field variables and their successive derivatives represents a challenge for any technique, given the strong mesh anisotropy.

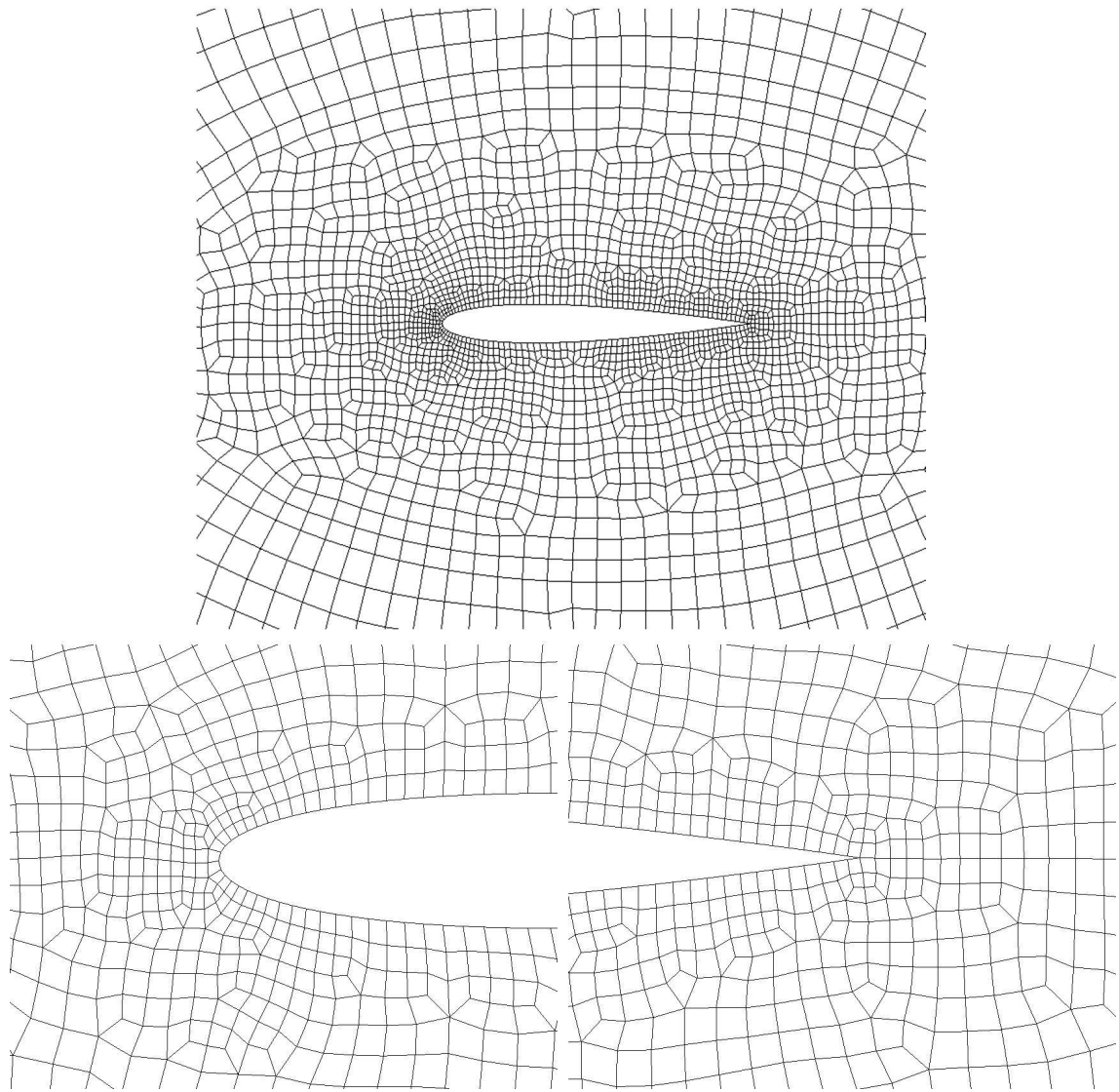


Fig. 7. Coarse grid for inviscid flow around a NACA 0012 aerofoil, 5322 cells.

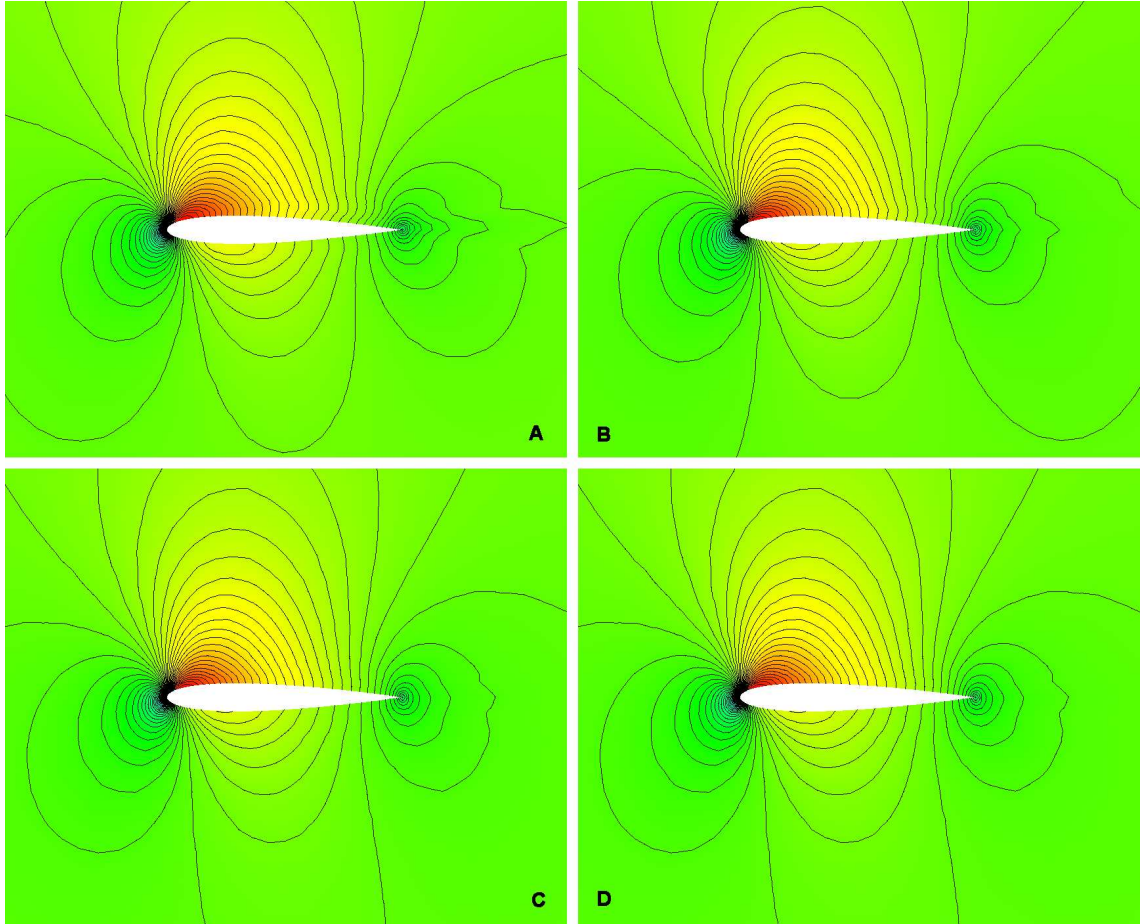


Fig. 8. Subsonic inviscid flow around a NACA 0012 aerofoil ($M = 0.63$, $\alpha = 2^\circ$): Mach number contours obtained with linear (A), quadratic (B) and cubic (with 2 and 3 Gauss points, C and D respectively) reconstructions.

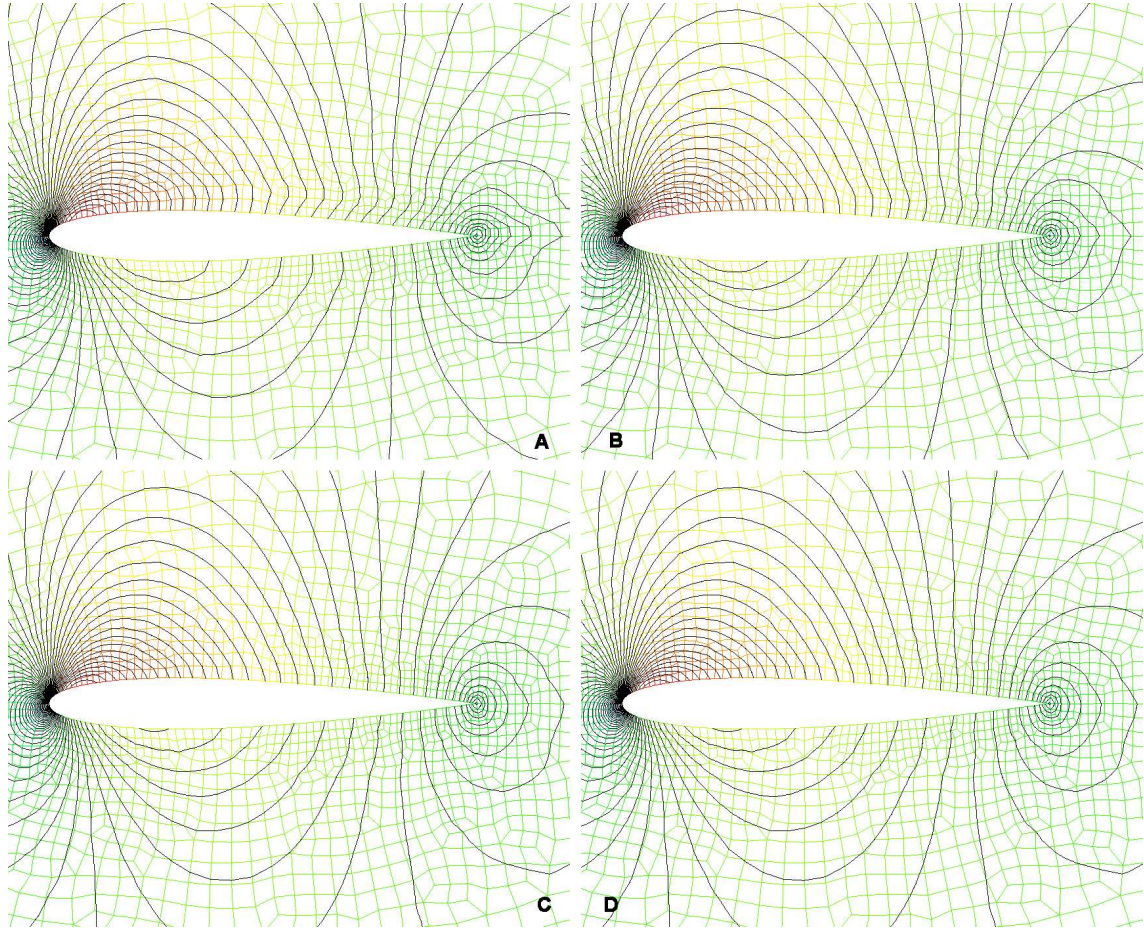


Fig. 9. Subsonic inviscid flow around a NACA 0012 aerofoil ($M = 0.63$, $\alpha = 2^\circ$): close-up view of the Mach number contours obtained with linear (A), quadratic (B) and cubic (with 2 and 3 Gauss points, C and D respectively) reconstructions.

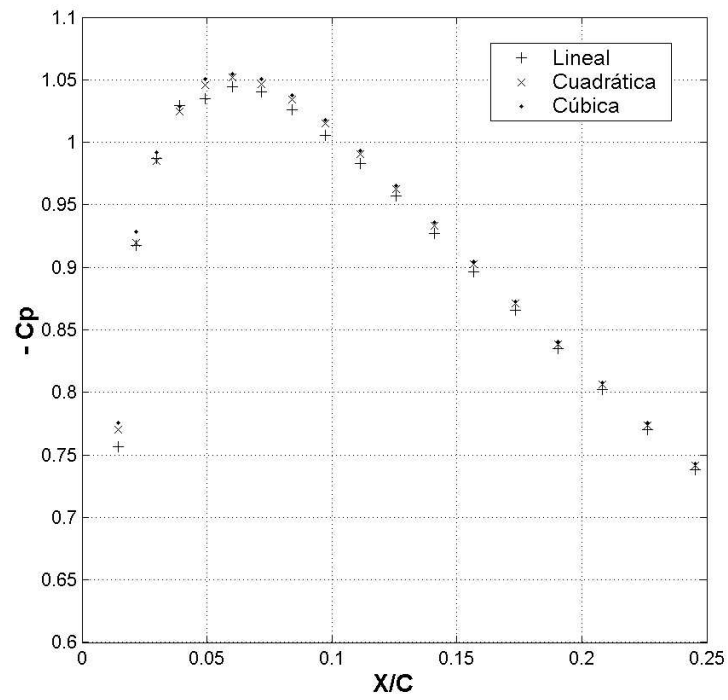


Fig. 10. Subsonic inviscid flow around a NACA 0012 aerofoil ($M = 0.63$, $\alpha = 2.^\circ$): surface pressure coefficients C_p (detail).

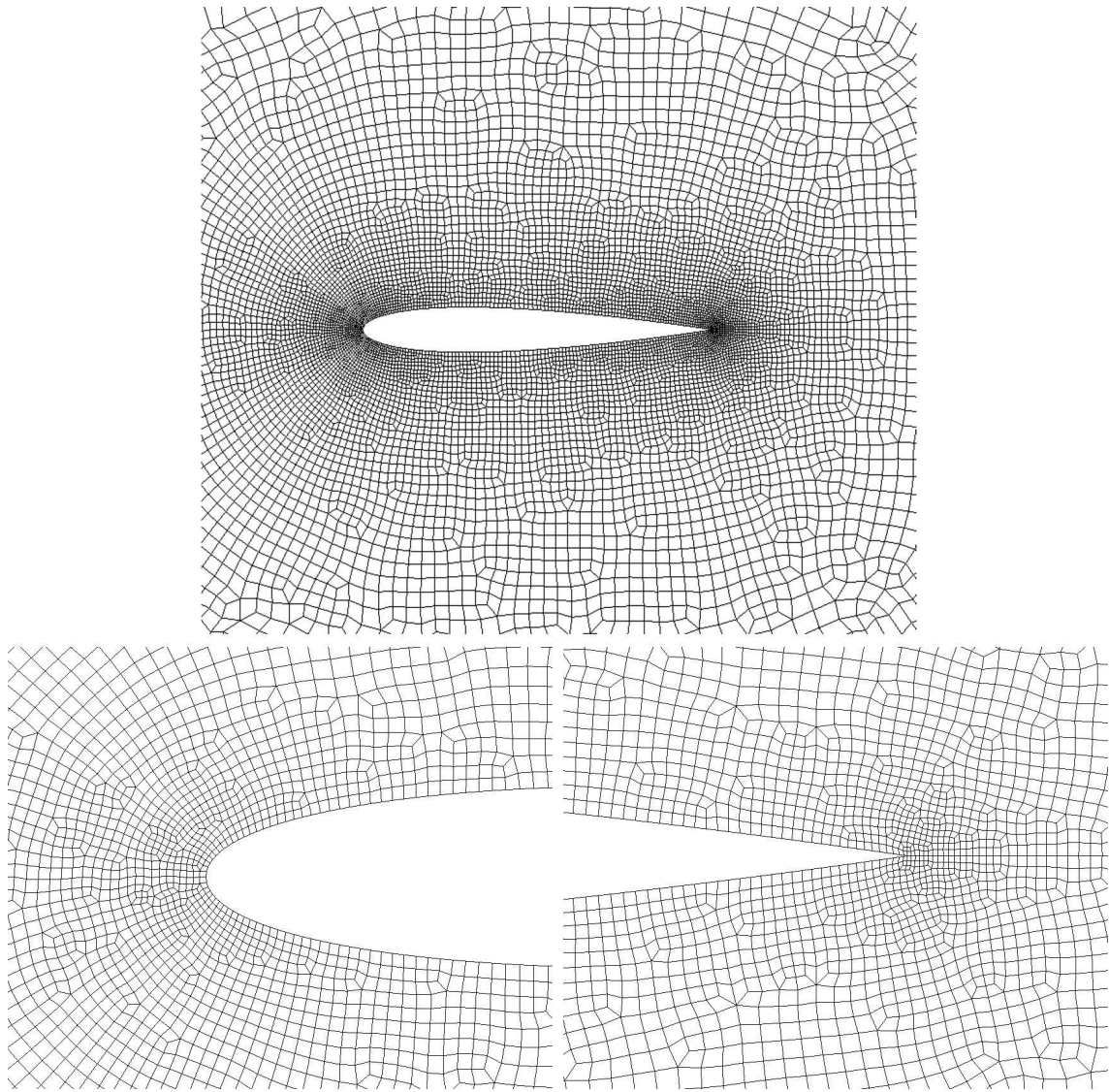


Fig. 11. Coarse grid for inviscid flow around a NACA 0012 aerofoil, 12243 cells.

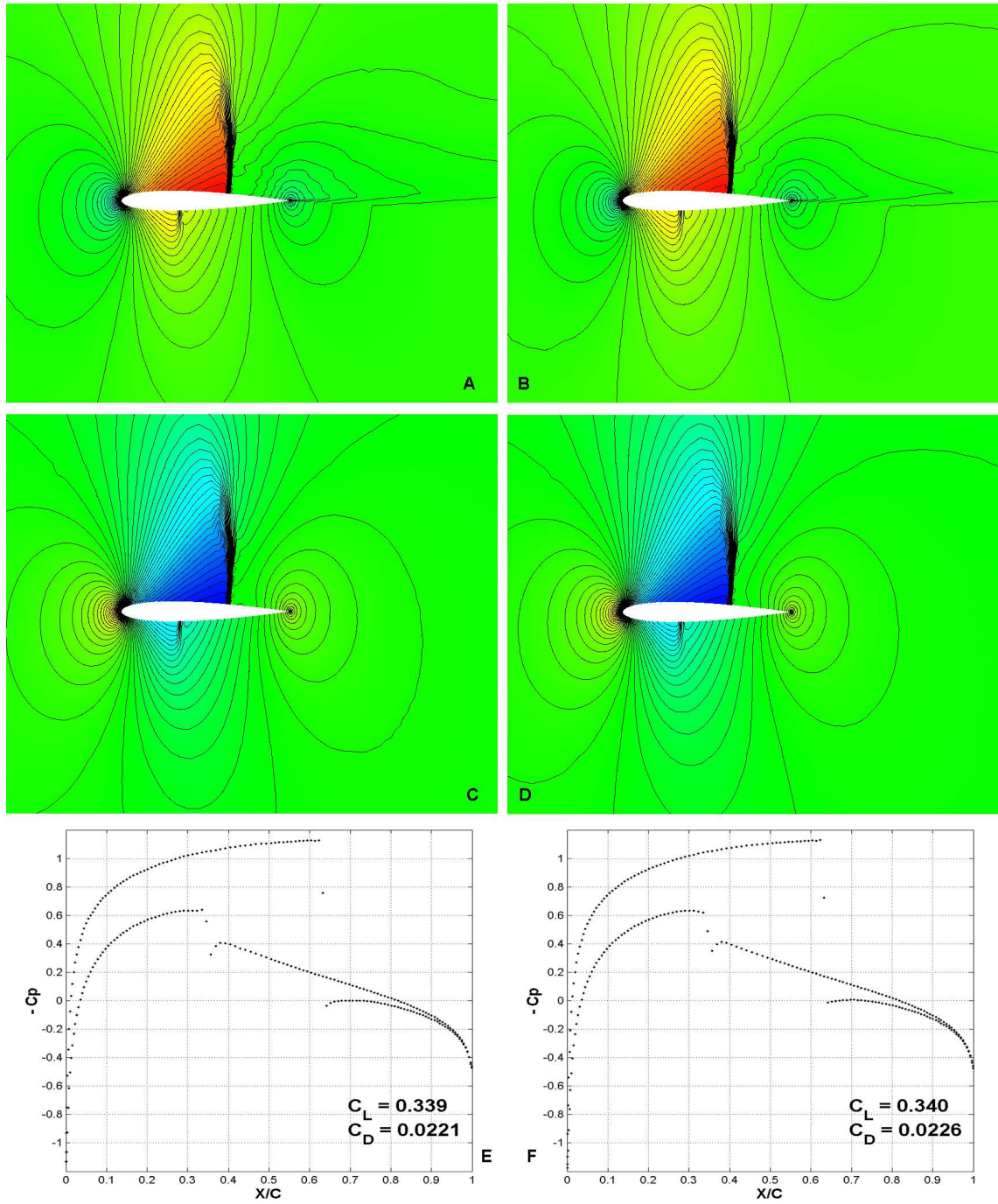


Fig. 12. Inviscid flow around a NACA 0012 aerofoil ($M = 0.8$, $\alpha = 1.25^\circ$): results obtained using *quadratic reconstruction* with either the BJ limiter (A-C-E) or the PC5 limiter (B-D-F). Mach number contours (A-B), pressure contours (C-D) and surface pressure coefficients C_p (E-F).

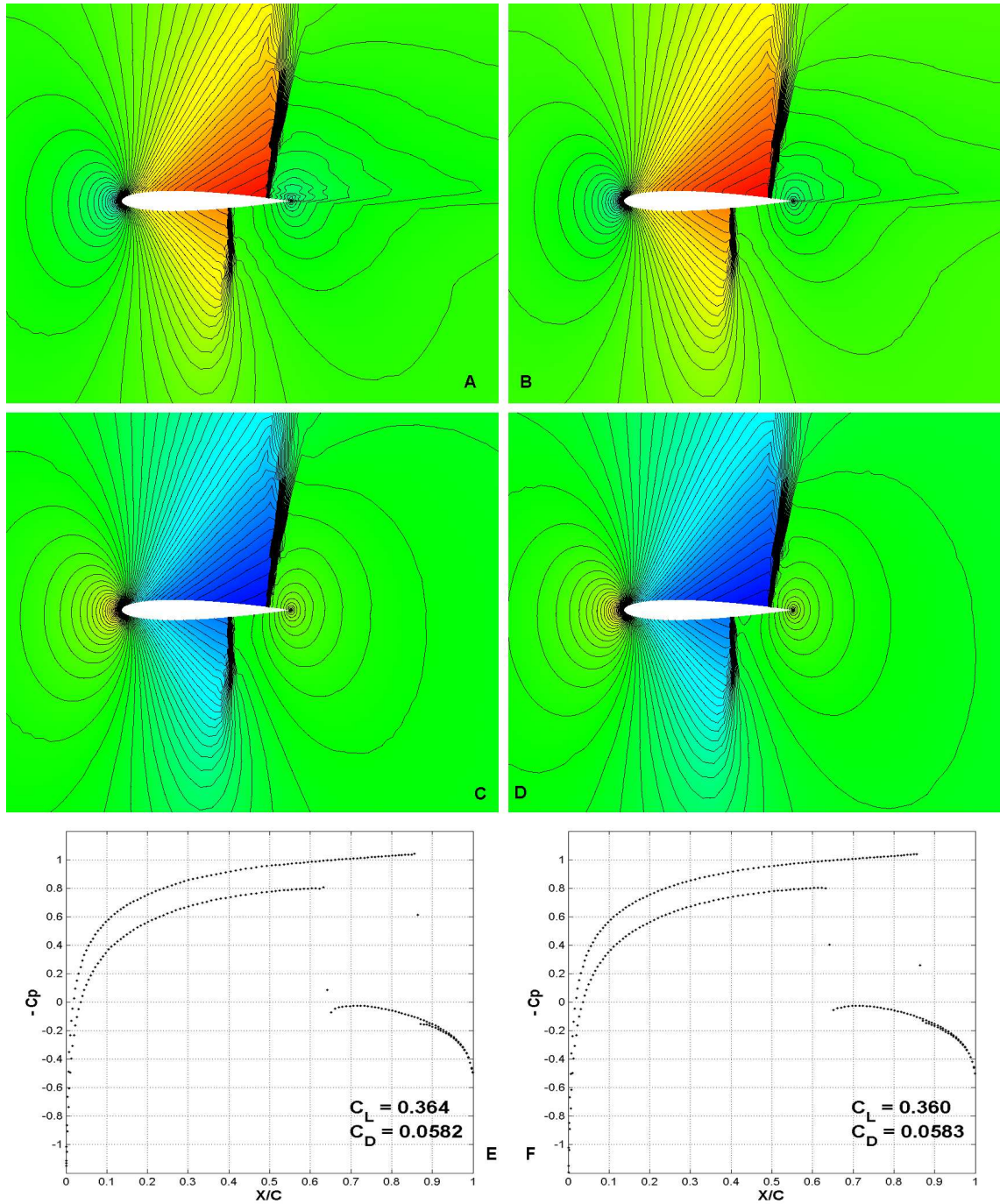


Fig. 13. Inviscid flow around a NACA 0012 aerofoil ($M = 0.85$, $\alpha = 1^\circ$): results obtained using *quadratic reconstruction* with either the BJ limiter (A-C-E) or the PC5 limiter (B-D-F). Mach number contours (A-B), pressure contours (C-D) and surface pressure coefficients C_p (E-F).

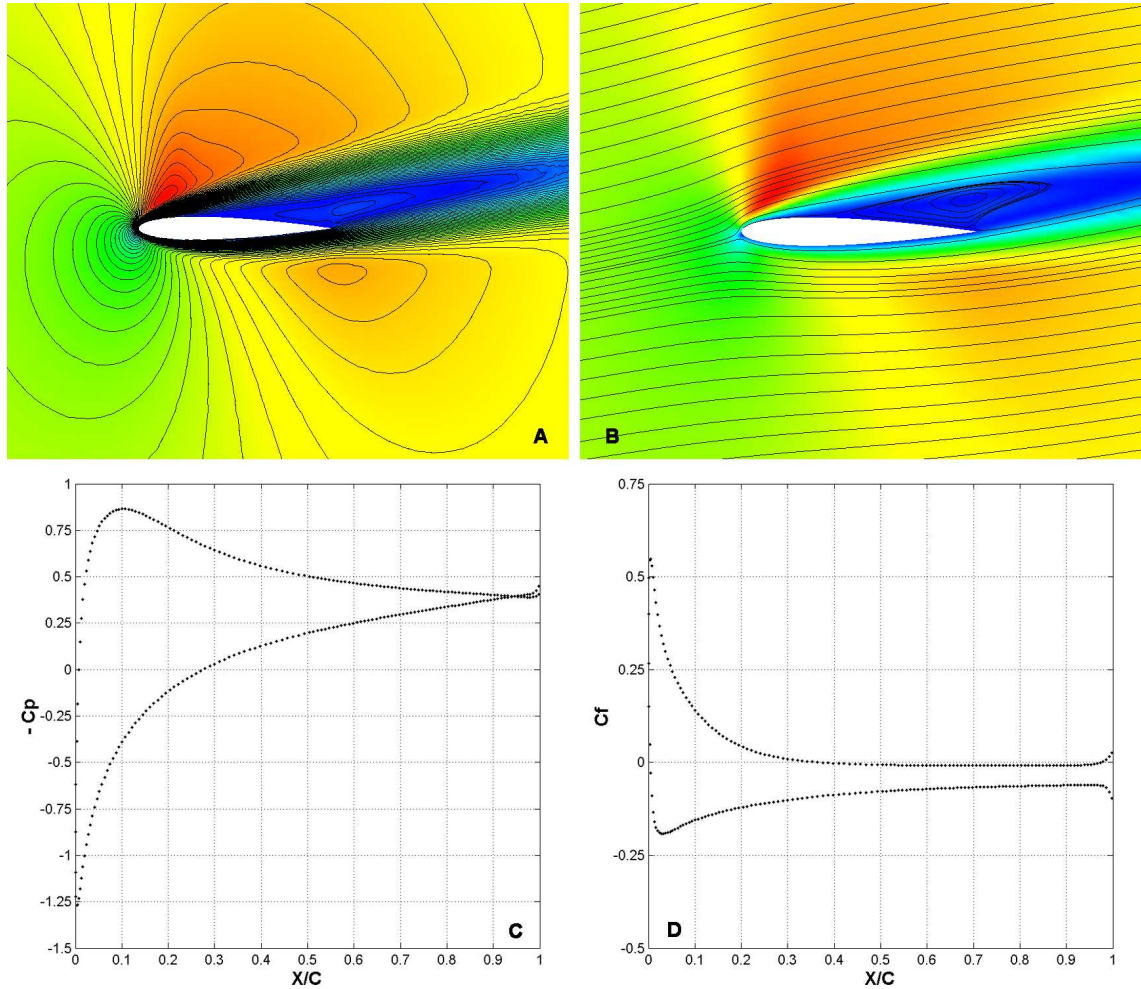


Fig. 14. Viscous flow around a NACA 0012 aerofoil ($M = 0.8$, $\alpha = 10^\circ$, $Re = 500$): Mach number contours (A), streamline patterns (B), surface pressure coefficients C_p (C) and skin friction coefficients C_f (D). Results computed using *quadratic reconstruction*.

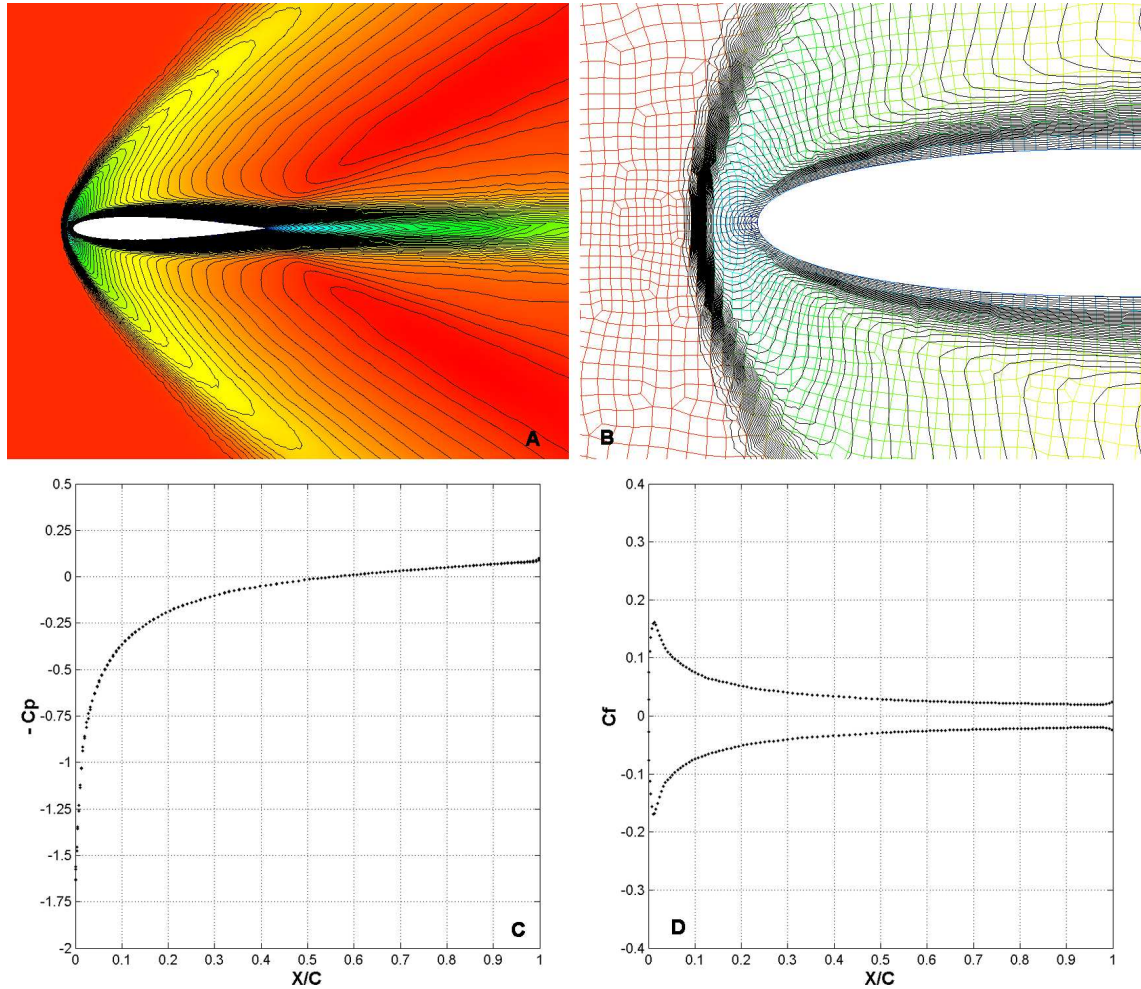


Fig. 15. Viscous flow around a NACA 0012 aerofoil ($M = 2$, $\alpha = 0^\circ$, $Re = 2000$): Mach number contours (A), close-up view of the Mach number contours near the leading edge (B), surface pressure coefficients C_p (C) and skin friction coefficients C_f (D). Results computed using *quadratic reconstruction* and the PC5 limiter.

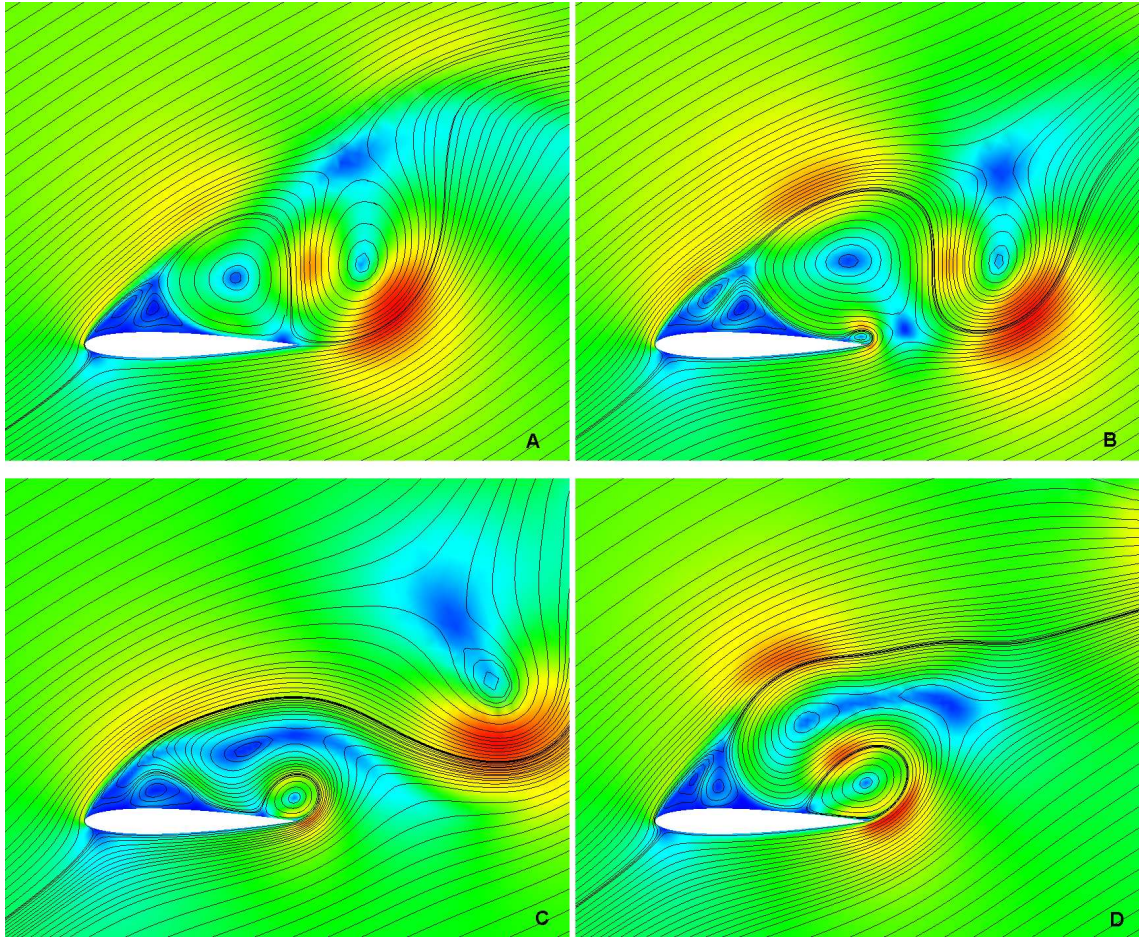


Fig. 16. Unsteady viscous flow around a NACA 0012 aerofoil ($M = 0.3$, $\alpha = 30^\circ$, $Re = 3000$): streamline patterns and Mach number field for non-dimensional times $t^* = 79.8$ (A), $t^* = 86.7$ (C), $t^* = 89.7$ (B) and $t^* = 94.1$ (D).

6 Acknowledgements

This work has been partially supported by the “Ministerio de Educación y Ciencia” of the Spanish Government (Grants DPI#2001-0556 and DPI#2004-05156) cofinanced with FEDER funds, the “Xunta de Galicia” (Grants # PGDIT01PXI11802PR and PGIDIT03PXIC118002PN) and the University of La Coruña.

Mr. Cueto-Felgueroso gratefully acknowledges the support received from “Fundación de la Ingeniería Civil de Galicia” and “Colegio de Ingenieros de Caminos, Canales y Puertos”. Part of this paper was written while Mr. Cueto-Felgueroso was visiting the University of Wales Swansea during the first semester of 2004. The support received from “Caixanova” and the kind hospitality offered by Prof. Javier Bonet and his research group are gratefully acknowledged.

References

- [1] S.K. Godunov. A difference method for the numerical calculation of discontinuous solutions of hydrodynamic equations. *Mat. Sbornik*. **47**(3):271-306 (1959).
- [2] A. Harten, P. Lax, B. Van Leer. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Rev.* **25**:35-61 (1983).
- [3] T.J. Barth, D.C. Jespersen. The design and application of upwind schemes on unstructured meshes. *AIAA-89-0366*(1989).
- [4] N.T. Frink. Upwind scheme for solving the Euler equations on unstructured tetrahedral meshes. *AIAA J.* **30**(1):70 (1992).
- [5] V. Venkatakrishnan. Convergence to steady state solutions of the Euler equations on unstructured grids with limiters. *Journal of Computational Physics*. **118**:120–130 (1995).
- [6] P. Jawahar, H. Kemath. A high-resolution procedure for Euler and Navier-Stokes computations on unstructured grids. *Journal of Computational Physics*. **164**:165–203 (2000).
- [7] T.J. Barth. Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations. *VKI Lecture Series 1994-05*.(1995).
- [8] J.M. Vaassen, P Wautelet, J.A. Essers. A quadratic reconstruction scheme for hypersonic reacting flows on unstructured meshes. *ECCOMAS Computational Fluid Dynamics Conference*. Swansea, Wales, UK (2001).
- [9] P. Lancaster, K. Salkauskas. Surfaces generated by moving least squares methods. *Math. of Comp.* **155**:141–158 (1981).
- [10] W.K. Liu, S. Li, T. Belytschko. Moving least-squares reproducing kernel methods: (I) Methodology and Convergence. *Computer Methods in Applied Mechanics and Engineering*. **143**:113–154 (1997).
- [11] S. Li, W.K. Liu. Moving least-squares reproducing kernel methods: (II) Fourier analysis. *Computer Methods in Applied Mechanics and Engineering*. **139**:159–193 (1996).
- [12] S. Li, W.K. Liu. Reproducing kernel hierarchical partition of unity, part I - Formulation and theory. *International Journal for Numerical Methods in Engineering*. **45**:251–288 (1999).
- [13] A. Gossler. Moving Least-Squares: a numerical differentiation method for irregularly spaced calculation points. SANDIA Report, SAND2001-1669 (2001).
- [14] D. Levin. The approximation power of moving least-squares. *Mathematics of Computation*. **67**(224):1517-1531 (1998).
- [15] P.L. Roe. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics*. **43**:357–372 (1981).

- [16] L. Cueto-Felgueroso, I. Colominas, J. Fe, F. Navarrina, M. Casteleiro. High order finite volume schemes on unstructured grids using Moving Least-Squares reconstruction. Application to shallow water dynamics. *International Journal for Numerical Methods in Engineering*. (in press) (2005).
- [17] L. Cueto-Felgueroso. Particles, finite volumes and unstructured grids: numerical simulation of fluid dynamics problems. PhD Thesis (in Spanish). Universidade da Coruña (2005).
- [18] L. Cueto-Felgueroso, I. Colominas, G. Mosqueira, F. Navarrina, M. Casteleiro. On the Galerkin formulation of the SPH method. *International Journal for Numerical Methods in Engineering*. **60**:1475–1512 (2004).
- [19] A. Huerta, Y. Vidal, P. Villon. Pseudo-divergence-free element free Galerkin method for incompressible fluid flow. *Computer Methods in Applied Mechanics and Engineering*. **193**:1119–1136 (2004).
- [20] F.M. White. Viscous fluid flow. (2nd edition) McGraw-Hill. New York (1991).
- [21] B. van Leer. Towards the ultimate conservative difference scheme V. A second order sequel to Godunov’s method. *Journal of Computational Physics*. **32**:101 (1979).
- [22] J. Van Rosendale. Floating shock fitting via Lagrangian adaptive meshes. *ICASE 94-89*(1989).
- [23] G. D. Van Albada, B. Van Leer, W. W. Roberts. A comparative study of computational methods in cosmic gas dynamics. *Astronomy and Astrophysics* **108**: 76 (1982).
- [24] C.-W. Shu, S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*. **77**:439-471 (1988).
- [25] J. Sarrate, A. Huerta. Efficient unstructured quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*. **49**:1327–1350 (2000).
- [26] P.S. Murthy, V.S. Holla, H. Kamath. Unsteady Navier-Stokes solutions for a NACA 0012 airfoil. *Computer Methods in Applied Mechanics and Engineering*. **186**:85-99 (2000).